Malnad College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

Hassan – 573202, Karnataka, India

Project (18EC802) Report on

"STROKE PREDICTION USING LOGISTIC REGRESSION"

is submitted to Malnad College of Engineering, Hassan, during the academic year 2021-22 in partial fulfillment for the award of the degree of

Bachelor of Engineering

in

Electronics & Communication Engineering

by

SHASHIDHAR GOWDA S (4MC18EC062) **SHREYAS B C (4MC18EC064)**

SPOORTHI K R (4MC18EC070) **SWATHI H AIRANI(4MC18EC074)**

Under the guidance of

Mrs. Swathi H Y
Assistant Professor
Dept. of ECE,
MCE, Hassan

Department of Electronics & Communication Engineering

Malnad College of Engineering

PB# 21, Hassan – 573 202, Karnataka, India

Tel:08172-245093 **Fax:08172-245683** **Website: www.mcehassan.ac.in**

2021-22

Department of Electronics & Communication Engineering

CERTIFICATE

Certified that the Project(18EC802) work titled

“STROKE PREDICTION USING LOGISTIC REGRESSION”

is a bonafide work carried out by

SHASHIDHAR GOWDA S (4MC18EC062)	**SHREYAS B C (4MC18EC064)**
SPOORTHI K R(4MC18EC070)	**SWATHI H AIRANI (4MC18EC074)**

in partial fulfilment for the award of

Bachelor of engineering in Electronics & Communication Engineering
of
Malnad College of Engineering,Hassan

affiliated to

Visvesvaraya Technological University, Belagavi

during the year 2021-22. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the project report deposited in the Department Library. The Project report has been approved, as it satisfies the academic requirements in respect of Project Work prescribed for the Bachelor of Engineering Degree.

Mrs. Swathi H Y	Dr. Srikanth P C	Dr. C V Venkatesh
Project Guide Assistant Professor Department of ECE MCE, Hassan	Head of the Dept. Department of ECE MCE, Hassan	Principal MCE, Hassan

External Viva

Name of the Examiners	Signature with Date
1.	
2.	

DECLARATION

We **SHASHIDHAR GOWDA S (4MC18EC062), SHREYAS B C(4MC18EC064), SPOORTHI K R (4MC18EC070), SWATHI H AIRANI (4MC18EC074),**
students of eighth semester, B.E Electronics and Communication Engineering of Malnad College of Engineering, Hassan, hereby declare that the project work titled " **STROKE PREDICTION USING LOGISTIC REGRESSION"** has been carried out by us under the guidance of **Mrs. SWATHI H Y**, Assistant Professor, Department of Electronics & Communication Engineering, Malnad College of Engineering, Hassan in the partial fulfillment of requirements for the award of degree in Bachelor of Engineering Electronics and Communication Engineering Malnad College of Engineering, Hassan under Visvesvaraya Technological University, Belgavi during the academic year 2021-22.

We also declare that, to the best of our knowledge and belief, the matter embodied in this project has not been submitted previously by us for the award of any degree or to any other University.

Place: Hassan

Date:

SHASHIDHAR GOWDA S (4MC18EC062)

SHREYAS B C (4MC18EC064)

SPOORTHI K R(4MC18EC070)

SWATHI H AIRANI (4MC18EC074)

ACKNOWLEDGEMENT

This project involved the collection and analysis of information from many people.

We are deeply indebted to our Project Guide **Mrs. Swathi H Y**, Assistant Professor, Department of Electronics and Communication Engineering, Malnad College of Engineering, Hassan for the valuable guidance, support and useful suggestions for this project work.

We also express our heartfelt gratitude to our Project Co-Ordinator **Mrs. Prakruthi H L,** Assistant Professor, Department of Electronics and Communication Engineering, Malnad College of Engineering for the boundless cooperation and encouragement.

We would like to express our sincere thanks to **Dr. PC Srikanth,** Professor and Headof the Department of Electronics and Communication Engineering, Malnad College of Engineering, Hassan for all the facilities provided in the department.

We take this opportunity to acknowledge our gratitude to **Dr. C V Venkatesh,** Principal of Malnad College of Engineering, Hassan for providing such an ambience in the premises which madethis endeavor a success.

Finally, we thank our teaching and non-teaching staffs of Electronics and Communication Engineering and classmates who helped us directly or indirectly to carry outthis project.

ABSTRACT

A Stroke is a health condition that causes damage by tearing the blood vessels in the brain. It can also occur when there is a blockage in the blood flow and other nutrients to the brain. According to the World Health Organization (WHO), stroke is the leading cause of death anddisability globally. Most of the work has been carried out on the prediction of heart stroke but very few show the risk of brain stroke.

With this thought, various machine learning models are built to predict the possibility of stroke in the brain. Various physiological factors are considered and machine learning algorithms like Logistic Regression, Decision Tree Classification, Random Forest Classification, K-Nearest Neighbors, Support Vector Machine are used to train five different models for accurate prediction. The algorithm that best performed this task is logistic regression w i t h a n accuracy of approximately 82%.

CHAPTER **Page no.**

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Introduction

According to the Centers for Disease Control and Prevention (CDC), stroke is the fifth-leading cause of death in the United States .Stroke is an non-communicable infection that is liable for around 11% of total deaths. Consistently, over 795,000 individuals in the United States experience the ill effects of a stroke. It is the fourth significant reason for death in India.

In 1970, the World Health Organization defined stroke as 'rapidly developed clinical signs of focal (or global) disturbance of cerebral function, lasting more than 24 hours or leading to death, with no apparent cause other than of vascular origin'. Central nervous system infarction is defined as brain, spinal cord, or retinal cell death attributable to ischemia, based on neuropathological, neuroimaging, and/or clinical evidence of permanent injury. Central nervous system infarction occurs over a clinical spectrum: Ischemic stroke specifically refers to central nervous system infarction accompanied by overt symptoms, while silent infarction by definition causes no known symptoms. Stroke also broadly includes intracerebral hemorrhage and subarachnoid hemorrhage. The updated definition of stroke incorporates clinical and tissue criteria and can be incorporated into practice, research, and assessments of the public health.

With the advancement of technology in the medical field, the occurrence of stroke can be predicted using Machine Learning Algorithms, which are constructive in making an accurate prediction and give correct analysis. Very less works have been performed on Brain stroke. The key components of the approaches used and results obtained are that among the five different classification algorithms used Logistic Regression has best performed obtaining a higher accuracy metric. The limitation with this model is that it is being trained on textual data and not on real time brain images. This shows the implementation of five Machine Learning classification algorithms.

A data set is chosen from Kaggle with various physiological traits as its attributes to proceed with this task. These traits are later analyzed and used for the final prediction. The dataset is initially cleaned and made ready for the machine learning model to understand. This step is called Data Preprocessing. For this, the dataset is checked for null values and fill them. Then Label encoding is performed to convert string values into integers followed by one-hot encoding, if necessary.

After Data Preprocessing, the dataset is split into training and testing data. A model is then built using this new data using various Classification Algorithms. Accuracy is calculated for all these algorithms and compared to get the best-trained model for prediction.

After training the model and calculating the accuracy, an HTML page and a flask application are developed. The web application is for the user to enter the values for prediction. The flask application is a framework that connects the trained model and the web application.

1.2 Literature Survey

Singh, M.S.,Choudhary et.al.[4], predicted stroke based on on Cardio vascular Health Study(CHS) data set using five machine learning techniques. As an optimal solution, the authors used a combination of the Decision Tree with the C4.5 algorithm, Principal Component Analysis, Artificial Neural Networks, and Support Vector Machine. But the CHS Dataset taken for this work had a smaller number of input parameters.

Pradeepa, S., et.al[5] , carried out stroke prediction from the social media posts posted by people. In this particular work, the authors have used the DRFS methodto find the various symptoms associated with stroke disease. The usage of Natural Language Processing to extract the text from the social media posts adds up to the overall execution timeof the model which is not desirable.

Vamsi Bandi et.al[6], the authors have performed the task of stroke prediction by using an improvised random forest algorithm. This was used to analyze the levels of risks obtained with the strokes. As suggested by the authors, this method is said to have performed better when compared to the existing algorithms. This particular research is limited to very few types of strokes and cannot be used for any new stroke type in the future.

Nwosu, C.S., et.al[7], Predicting stroke from electronic health records. In: 41st Annual International Conference of the IEEE Engineeringin Medicine and Biology Society IEEE (2019). The calculated accuracy for Decision Tree was 74.31%, Random Forest was 74.53%, and multi-layer perceptron was 75.02%. This paper suggests that Multi-layer perceptron is more accurate than the other two methods. Accuracy score was the only metric used for calculating the performance that might not always give favorable results.

Research carried out in Fahd Saleh Alotaibi: Implementation of Machine Learning Model to Predict Heart Failure Disease. In: International Journal of Advanced Computer Science and

Applications (IJACSA) (2019). [8] shows the implementation of machine learning model to predict heart stroke. They used various machine learning techniques like Decision tree, Naïve Bayes, SVM to build the model and later compared their performance. They obtained a maximum accuracy of 60% from the used algorithms which is pretty less.

Ohoud Almadani, et.al[9], Prediction of Stroke using Data Mining Classification Techniques. In: International Journal of Advanced Computer Science and Applications (IJACSA) (2018). [9], the authors have used different data mining classification techniques to predict the possibility of a stroke. The dataset was taken from the Ministry of National Guards Health Affairs Hospitals, Kingdom of Saudi Arabia. The three classification algorithms used were C4.5, Jrip and Multi layers perceptron (MLP). With these algorithms,the model obtained an accuracy of around 95%. Even though the paper claims to obtain an accuracy of 95%, the time taken for training and predicting is higher as the authors have useda combination of complex algorithms.

Research carried out by Kansadub, et.al[10], suggests the usage of three differentalgorithms to predict the possibility of stroke. These algorithms are Naïve Bayes, Decision Tree, and Neural Networks. This paper concluded that the Decision tree has the highest accuracy (about 75%) of the other two algorithms. But this model could not suit the real- world examples based on the values obtained from the confusion matrix

1.3 Problem Statement

Stroke is the second leading cause of death worldwide and remains an important health burden both for the individuals and for the national healthcare systems. Potentially modifiable risk factors for stroke include hypertension, cardiac disease, diabetes, and dysregulation of glucose metabolism, atrial fibrillation and lifestyle factors. Therefore, the goal of our project is to apply machine learning algorithm over large existing data sets to effectively predict the stroke based on potentially modifiablerisk factors. Then it is intended to develop the application to provide a personalized warning on the basis of each user’s level of stroke risk and a lifestyle correction message about the stroke risk factors.

1.4 Proposed System

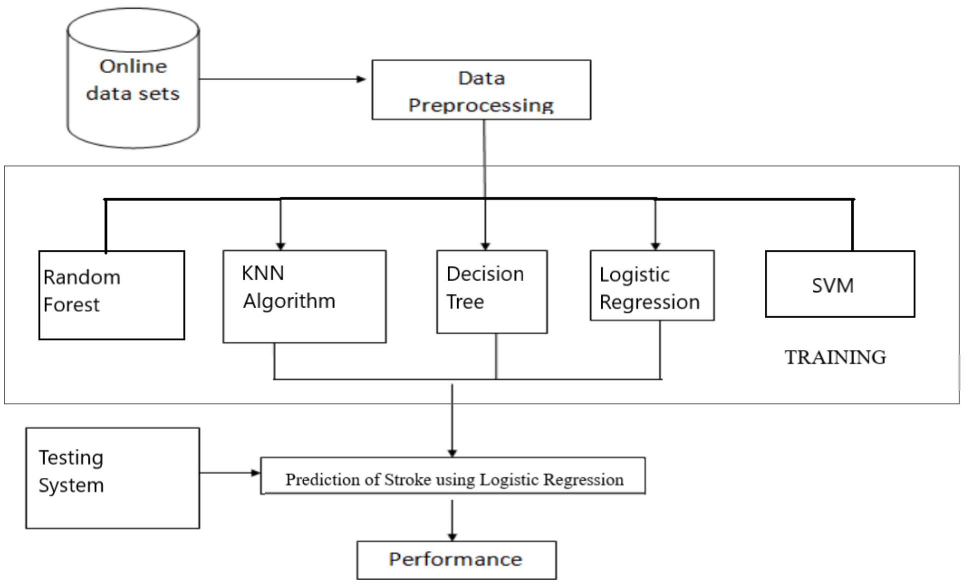


Fig 1.1. Proposed System

Dataset

This dataset is used to predict whether a patient is likely to get stroke based on the input parameterslike gender, age, various diseases, and smoking status. Each row in the data provides relavant information about a patient.

Attributes Information

The data contains 5110 observations with 12 attributes.

• id: unique identifier.

• gender: "Male", "Female" or "Other".

• age: age of the patient.

• hypertension: hypertension means high blood pressure. 0 if the patient doesn't have hypertension,1 if the patient has hypertension.

• heart_disease: 0 if the patient doesn't have any heart diseases, 1 if the patient has a heart disease

• ever_married: "No" or "Yes".

• work_type: "children", "Govt_job", "Never_worked", "Private" or "Self-employed".

• Residence_type: "Rural" or "Urban".

- avg_glucose_level:average glucose level in blood.
- bmi: body mass index, As a measure of obesity.
- smoking_status: "formerly smoked", "never smoked", "smokes" or "Unknown"*.
- stroke: 1 if the patient had a stroke or 0 if not.

*Note: "Unknown" in smoking_status means that the information is unavailable for this patient.

Data pre-processing

Data pre-processing is a primary step for adequately describing the data for the machine learning algorithm. It is playing an essential role in improving the performance results of machine learning. In this stage, several steps are applied. Smoking-status and bmi features have many missing values. Mean is applied to fill missing values. Converting categorical features into numerical data using Label Encoder. The database is imbalanced data. Imbalanced data means there is an unbalanced ratio of values for each class label. We handle imbalanced data using randomresample techniques.

Training

Algorithms Involved-Few methodologies used in our projects are:

1. Decision Tree

Decision Tree classification is to solve both Regression and classification problems. This algorithm is a supervised learning method wherein the input variables already have their corresponding output variable. It is a tree-like structure. In this algorithm, the data continuously splits according to a particular parameter. A decision tree has two parts: Decision Node and Leaf node. The data splits at the former node, and the latter is the node that gives the outcome.

2. Random Forest

Random Forests are composed of multiple independent decision trees trained independently on a random subset of data. These trees are generated at the time of training, and the outputs are obtained from each decision tree. For the final prediction from this algorithm, a method called "voting" takes place. This method means that each decision tree votes for an output class (in this case, the two classes are: 'stroke' and 'no stroke'). The random forest chooses the class with the maximum number of votes as the final prediction.

3. Logistic Regression

Logistic Regression is a supervised learning algorithm used for predicting the probability of the output variable. This algorithm is the best fit when the output variable has binary values (0 or 1). As the output attribute in the dataset has only two possible values

4. Support Vector Machine

It is a supervised learning technique that can be associated with learning algorithms to analyze the data for both classification and regression. Support Vector Machine (SVM) scales relatively well to high dimensional data. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors.

5. KNN Algorithm

K-nearest neighbors classification: Another algorithm used for classification is K-Nearest Neighbors (KNN). It is also a supervised learning technique. It stores the dataset, and at the time of classification, it acts on the dataset. The working principle of KNN is to find Similarities between the new case (or data) and available data and then map the new case into thecategory that is most similar to the available categories.

Testing

The web page is built using simple HTML code. This application has an input form that will take the entered input values from the user to predict the occurrence of stroke. When the user clicks on the 'Check Here' button, the entered parameters are given to the flask application. A part of the HTML page is shown.The flask application which is basically a python code is the bridge between the web page and the trained machine learning model. The input values are sent to the flask application which in sends the values to the model for prediction**.** Once the machine learning model gets the input parameters, it predicts the output. The obtained result then is reflected back on the web page through the flask application for the user to see the prediction.

Prediction

For evaluating the performance of models, we have used the confusion matrix to calculate accuracy, precision, recall, and f-measure. Confusion matrix describes performance of a model on a set of test data. It gives two types of correct predictions and two types of incorrect predictions for the classifier. Table 2 shows the confusion matrix. TP is the predicted output as true positive, TN is the predicted output as true negative, FP is the predicted output as false positive, and FN is the predicted output as a false negative. The accuracy, precision, recall, and f-measure (f-score) are defined in the following

	Predicted Class 0	Predicted Class 1
Actual Class 0	TP	FN
Actual Class 1	FP	TN

Accuracy = **Number of correct predictions Total number of predictions**

Precision is the total number of correctly classified positive divide on the total number of predicted positive examples.

F-measure is a measurement that represents the relationship between Precision and Recall. F-Measure will always be nearer to the smaller value of Precision or Recall.

The recall is calculated as the ratio between the numbers of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.

Performance

Out of Random Forest, Decision Tree, KNN, Logistic Regression, SVM algorithms chosen, Logistic regression algorithm performs best with an accuracy of 93.835% , precision of 75%, recall of 90%, f-measure of 76%.

1.5 Organization Of The Report

The project is organized as follows. Chapter 1 gives Introduction, Literary Survey, Problem Statement, The Proposed System and Organization of the report. Chapter 2 explains about the Methodology, Block Diagram, Proposed Algorithm. Chapter 3 explains about the software requirements. Chapter 4 explains about final results that we have obtained Chapter 5 gives the description of Conclusion and Future Scope.

CHAPTER 2

System Specification And Overview

2.1 Methodology:

The following block diagram depicts the methodology involved in machine learning techniques.

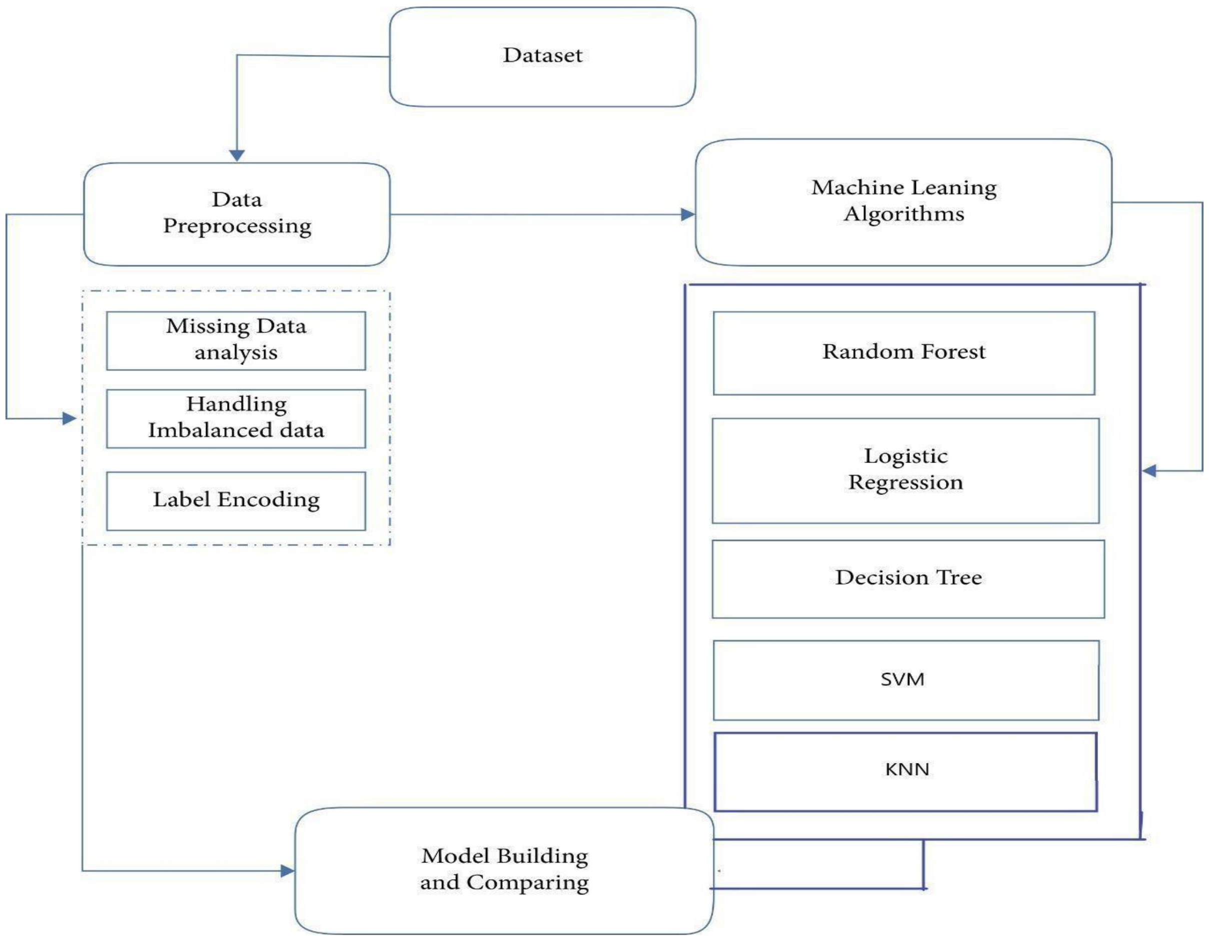


Fig 3.1. Flow chart of Methodology

Different datasets were considered from Kaggle. Out of all the existing datasets, anappropriate dataset was collected for model building. After collecting the dataset, next step lies in preparing the dataset to make the data moreclear and easily understood by the machine. This step is called as Data preprocessing.

This step includes handling of missing values, handling imbalanced data and performinglabel encoding that are specific for this particular dataset. Now that the data is preprocessed, it is ready for model building. For model building, preprocessed dataset along with machine learning algorithms are required. Logistic Regression, Decision Tree Classification algorithm, Random Forest Classification algorithm, K-Nearest Neighbor algorithm, Support Vector Classification algorithm are used. After building five different models, they are compared using five accuracy metricsnamely Accuracy Score, Precision Score, Recall Score, F1 Score. The comparison of the models gives the best model in terms of the accuracy metrics to proceed with the deployment phase. For deploying the model, an HTML page is developedto make it user-friendly for the user to enter the input parameters and get the result. The parameters entered by the user are sent to the model using a flask application which is basically a python framework that links the web application and the model together. The model takes the input parameters, predicts the output and returns the result to the flaskapplication. Now this flask will display that result on the web page for the user to check the result.

2.1Block diagram

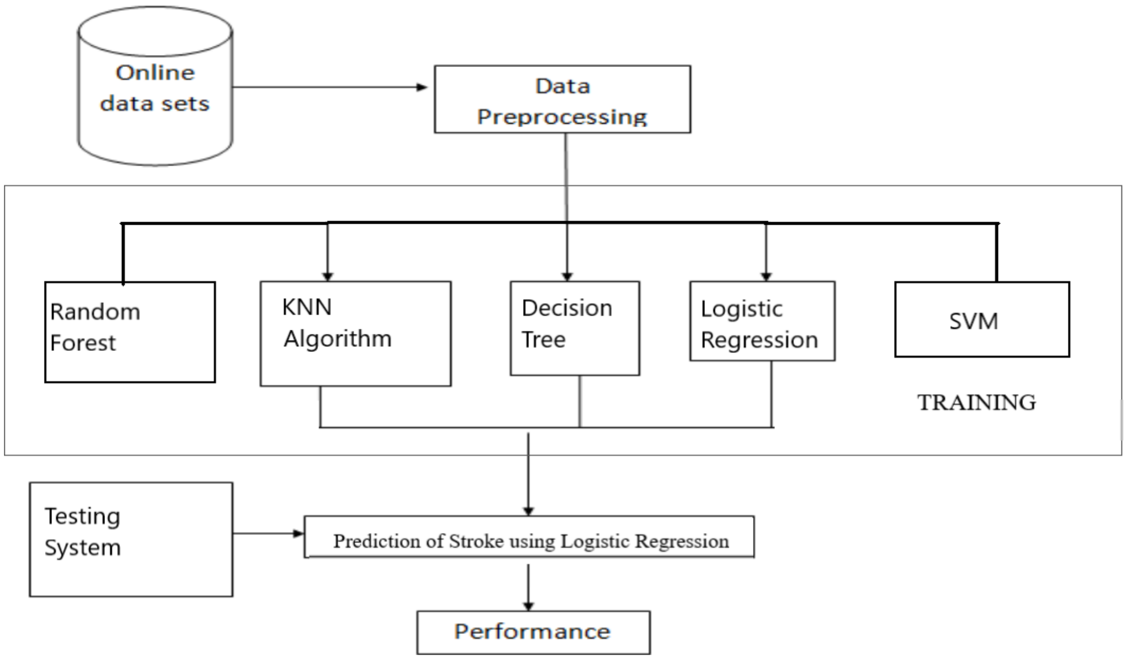


Fig3.2. Model algorithm

Data Set:

The stroke prediction Kaggle dataset was used to perform the study. There were 5110 rows and 12 columns in this dataset. The value of the output column stroke is either 1 or 0. The number 0 indicates that no stroke risk was identified, while the value 1 indicates that a stroke risk was detected. The probability of 0 in the output column (stroke) exceeds the possibility of 1 in the same column in this dataset. 249 rows in the stroke column have the value 1, whereas 4861 rows have the value 0. To improve accuracy, data preprocessing is used to balance the data.

Data-Preprocessing:

Before building a model, data preprocessing is done to remove unwanted noise andoutliers from the dataset that could lead the model to depart from its intended training. This stage addresses everything that prevents the model from functioning more efficiently. Following thecollection of the relevant dataset, the data must be cleaned and prepared for model development.As stated before, the dataset used has twelve characteristics. To begin with, the column id is omitted since its presence has no bearing on model construction. The dataset is then inspected for null values and filled if any are detected. The null values in the column BMI are filled usingthe data column’s mean in this case.

Label encoding converts the dataset’s string literals to integer values that the computer can comprehend. As the computer is frequently trained on numbers, the strings must be converted to integers. The gathered dataset has five columns of the data type string. All strings are encoded during label encoding, and the whole dataset is transformed into a collection of numbers**.**

2.3 Proposed Algorithms

The most common disease identified in the medical field is stroke, which is increasing year after year. Using the publicly accessible stroke prediction dataset, the study measured four commonly used machine learning methods for predicting brain stroke recurrence, which are asfollows:

(i) Random forest

Random forest is based on the theory of ensemble learning. It is a procedure that indulges various classifiers in resolving compounded drawbacks by enhancing the execution of this technology. Considering the average in enhancing in identifying the speed of the particular dataset by the random forest which consisting of multiple decision trees relying on different subset among the given dataset. Alternately rather than depending on a single decision tree, the random forest concludes every single tree and on the maximum identifications by finally displaying the identified outcome. The importance of each feature of a decision tree can be calculated as follows:

$$fi_i = \frac{\sum_{j:node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k \in all\ nodes} ni_k}$$

(ii) Decision tree

The decision tree consists of two apexes, those are Decision apex and the leaflet apex. Among these, the Decision apex shows the features of the number of limbs attached to it and are required in making decisions, on the other hand, the leaflet apexes represent the outcomes of these decision limbs and they do not have any limbs attached to them. The selection or implementation depends upon the type of dataset that is provided. Depending upon the requiredscenarios these are represented in the form of a graph by drawing all the possible outcomes based on the selection/complication. Since it starts with the source apex and diverges in variousdirections by developing a structure of the tree is calculated as:

$$\text{Gini Index (G)} = \sum_{i=1}^{c} P_i (1 - P_i)$$

(iii) Support Vector Machine

The main aim of this model is to develop the exact linear way or deterministic partition which separates n-proportional space into groups such that providing easy access of combining the data which is newly formed into their respective modules for further references. This type of sorting the data by an exact linearway can also be referred to as hyperplane. Since considering these outermost vector points that are supportive in building the hyperplane are termed as support points and so the algorithm is named as Support vector algorithm. With the help of the demo graphs that are categorized into two variant ways which are divided considering the deterministic partition or a hyperplane. The linear SVM is required in linearly differentiating the information/data, that represents dividing the dataset into two different classes by a unique linear separation. Data is termed to as the linear differential and the classifier is written as:

$$\textbf{Class 1}\ \text{(Low risk)} = (\mathrm{W} * \mathrm{X} + \mathrm{b}) \geq 1, \forall \mathrm{X}$$

$$\textbf{Class 2}\ \text{(High risk)} = (\mathrm{W} * \mathrm{X} + \mathrm{b}) \leq -1, \forall \mathrm{X}$$

(iv) Logistic-regression

Logistic regression comes under the technique of supervised learning, which is used for analyzing the absolute dependent values by making use of the variable among the required blocks of the independent values. Analysis of the output values can be determined by the logistic regression of the absolute dependent values. Hence the solutions can be drawn as the absolute or the differential variables. It may be of any form either numerical or binary variables i.e., Yes or No, 0 or 1, true or false, etc. In the computer-determined language, the values are given in the 0 or a format but this model represents the feasible value that lies between 0 and 1. The usage of the values is the only difference between Logistic regression and Linear Regression. The retrogradation of the problems can be settled using linear regression whereas the categorization of the issues was carried out by the Logistic regressions is written as:

$$\textit{Sigmoid Function}\ \emptyset(z) = \frac{1}{1+e^{-z}}$$

(v) KNN

The k-nearest neighbor classifier fundamentally relies on a distance metric. The better that metric reflects label similarity, the better the classified will be. The most common choice is the Minkowski distance.

$$\mathbf{dist}(\mathbf{x}, \mathbf{z}) = \left(\sum_{r=1}^{d} |x_r - z_r|^p \right)^{1/p}$$

CHAPTER 3

System Design

Software requirement

The Jupyter Notebook:

It is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Its uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Main features of the jupyter notebook are,

In-browser editing for code, with automatic syntax highlighting, indentation, and tab completion/introspection.

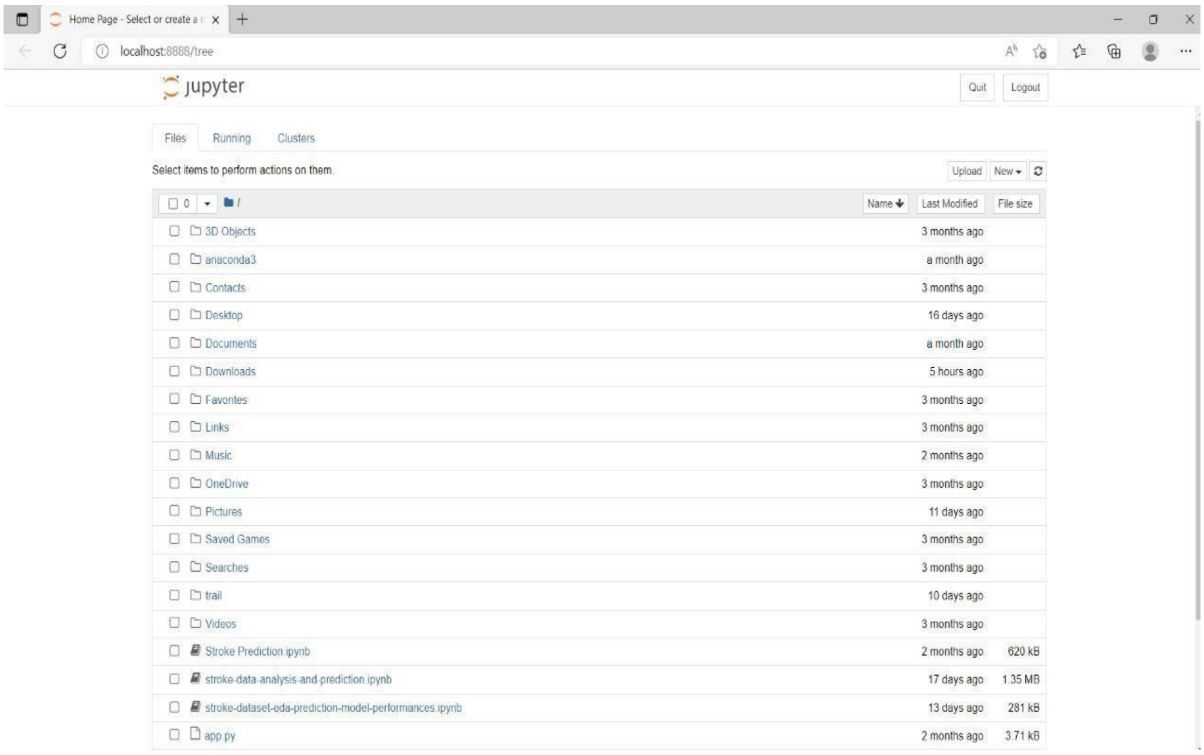
- The ability to execute code from the browser, with the results of computations attached to thecode which generated them.
- Displaying the result of computation using rich media representations, such as HTML, LaTeX, PNG, SVG, etc. For example, publication-quality figures rendered by the matplotlib library, canbe included inline.
- In-browser editing for rich text using the Markdown markup language, which can providecommentary for the code, is not limited to plain text.
- The ability to easily include mathematical notation within markdown cells using LaTeX, andrendered natively by MathJax.

Jupyter Notebook can connect to many *kernels* to allow programming in different languages. A Jupyter kernel is a program responsible for handling various types of requests (code execution, code completions, inspection), and providing a reply. Kernels talk to the other components of Jupyter using ZeroMQ, and thus can be on the same or remote machines. Unlike many other Notebook-like interfaces, in Jupyter, kernels are not aware that they are attached to a specific document, and can be connected to many clients at once. Usually kernels allow execution of only a single language, but there are a couple of exceptions. By default Jupyter Notebook ships with the IPython kernel. As of the 2.3 release (October 2014), there are 49 Jupyter -compatible kernels for many programming languages, including Python, R, Julia and Haskell.

A Jupyter Notebook can be converted to a number of open standard output formats (HTML, presentation slides, LaTeX, PDF, ReStructuredText, Markdown, Python) through "Download As" in the web interface, via the nbconvert library or "jupyter nbconvert" commandline interface in a shell. To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library is provided as a service through NbViewer which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

The notebook interface was added to IPython in the 0.12 release (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 is Jupyter 1.0). Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

JupyterLab is a newer user interface for Project Jupyter. It offers the building blocks of the classic Jupyter Notebook (notebook, terminal, text editor, file browser, rich outputs, etc.) in a flexible user interface. The first stable release was announced on February 20, 2018.



Select items to perform actions on them.

Name	Last Modified	File size
3D Objects	3 months ago	
anaconda3	a month ago	
Contacts	3 months ago	
Desktop	16 days ago	
Documents	a month ago	
Downloads	5 hours ago	
Favorites	3 months ago	
Links	3 months ago	
Music	2 months ago	
OneDrive	3 months ago	
Pictures	11 days ago	
Saved Games	3 months ago	
Searches	3 months ago	
trail	10 days ago	
Videos	3 months ago	
Stroke Prediction.ipynb	2 months ago	620 kB
stroke-data-analysis-and-prediction.ipynb	17 days ago	1.35 MB
stroke-dataset-eda-prediction-model-performances.ipynb	13 days ago	281 kB
app.py	2 months ago	3.71 kB

Fig 4.1 Getting started with jupyter

The libraries that we have imported here for this project are

- pandas
- matplotlib. pyplot
- scikit-Learn (Sklearn)
- seaborn
- pandas_ profiling
- pickle

Pandas is a software library written for the Python programming language for data manipulation and analysis. Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns calleddata frame that looks very similar to table in a statistical software (such as Excel).

matplotlib. pyplot is a state-based interface to matplotlib. It provides an implicit, MATLAB-like, way of plotting. It also opens figures on your screen, and acts as the figure GUI manager.

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis. Seaborn works easily with dataframes and the Pandas library. The graphs created can also be customized easily.

pandas_ profiling features a method to create a suite of Expectations based on the results of your Profile Report, which you can store, and use to validate another (or future) dataset.

The pickle module implements a fundamental, but powerful algorithm for serializing and de-serializing a Python object structure.
Pickle model provides the following functions –

- pickle. dump to serialize an object hierarchy, you simply use dump().
- pickle. load to deserialize a data stream, you call the loads() function.

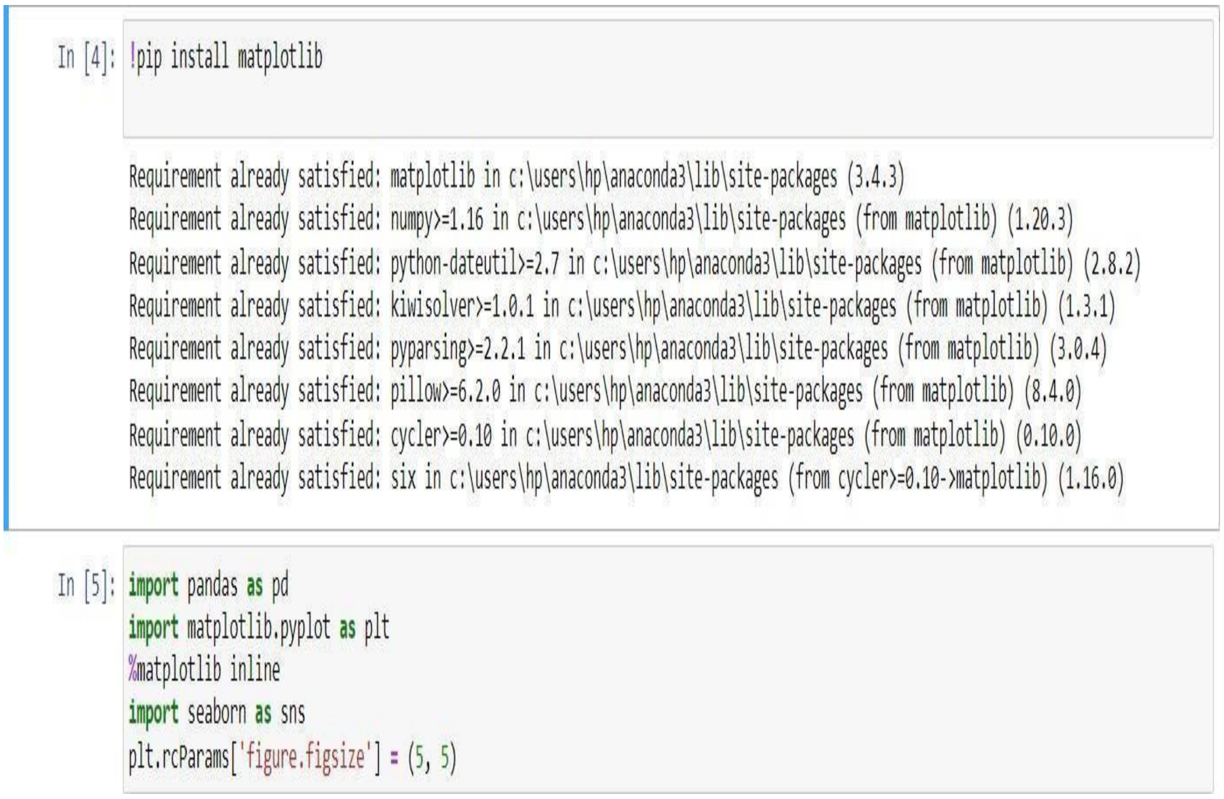
CHAPTER 4

Results And Discussions

4.1 Importing the required libraries:

The libraries that we have imported here for this project are

- pandas
- matplotlib.pyplot
- scikit-Learn (Sklearn)
- seaborn
- pandas_profiling
- pickle

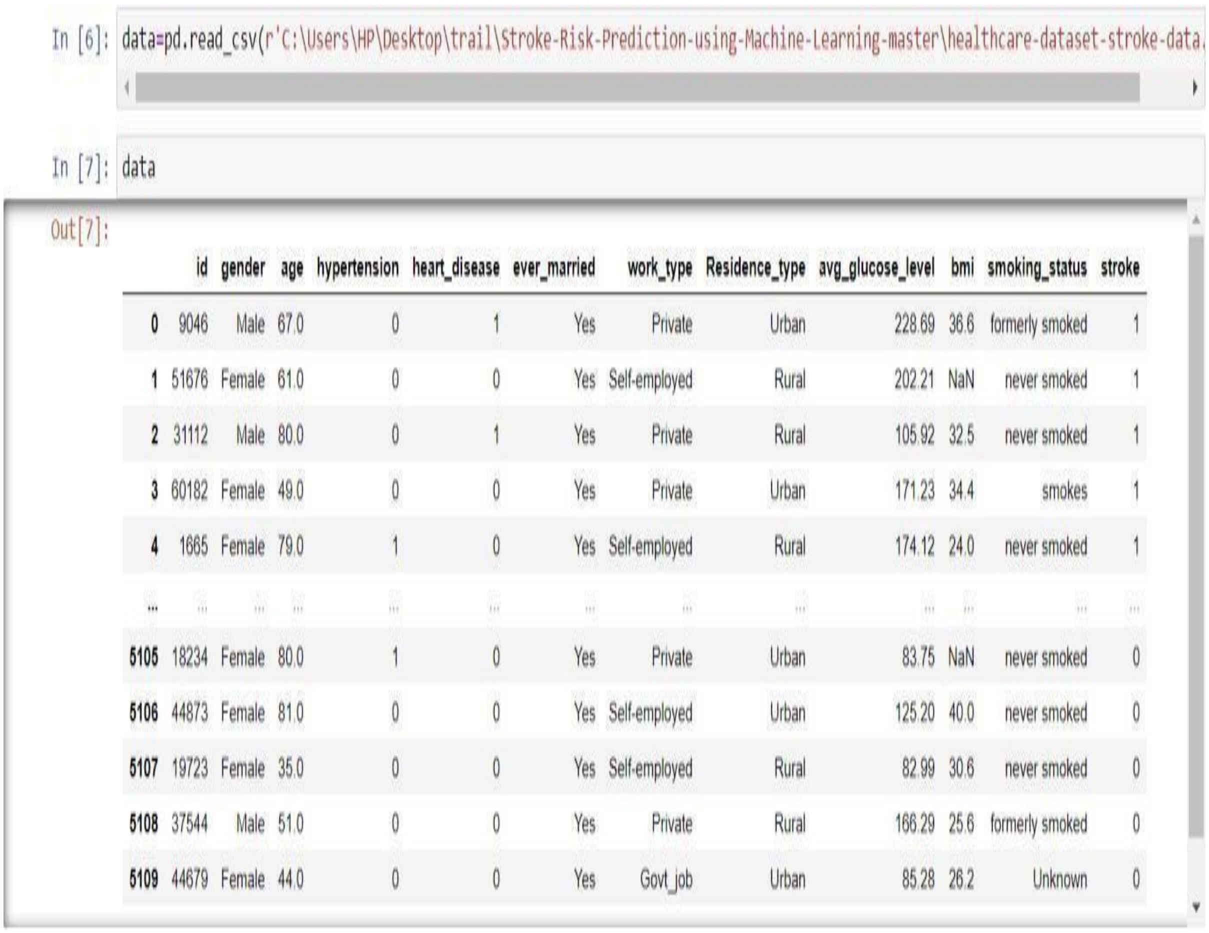


```
In [4]: !pip install matplotlib

Requirement already satisfied: matplotlib in c:\users\hp\anaconda3\lib\site-packages (3.4.3)
Requirement already satisfied: numpy>=1.16 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.20.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (1.3.1)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (8.4.0)
Requirement already satisfied: cycler>=0.10 in c:\users\hp\anaconda3\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: six in c:\users\hp\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib) (1.16.0)

In [5]: import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline
        import seaborn as sns
        plt.rcParams['figure.figsize'] = (5, 5)
```

Fig4.1 Importing the required libraries

4.2 Extracting and reading the dataset

The stroke prediction dataset was used to perform the study. There were 5110 rows and 12 columns in this dataset. The value of the output column stroke is either 1 or 0.
Data extraction involves pulling data from different sources and converting it into a useful format for further processing or analysis.CSV file refers to comma- separated values' file that is used to store data in a tabular format, similar to a spreadsheet. Each line in the file is an observation (or record), and each record has one or more features separated by commas. read_csv() function from pandas is used to deal with common problems when importing data.
Some of the features considered are gender, age, hypertension, bmi, heart_disease, work_type, avg_glucose_level, smoking_status, , Residence_type, ever_married

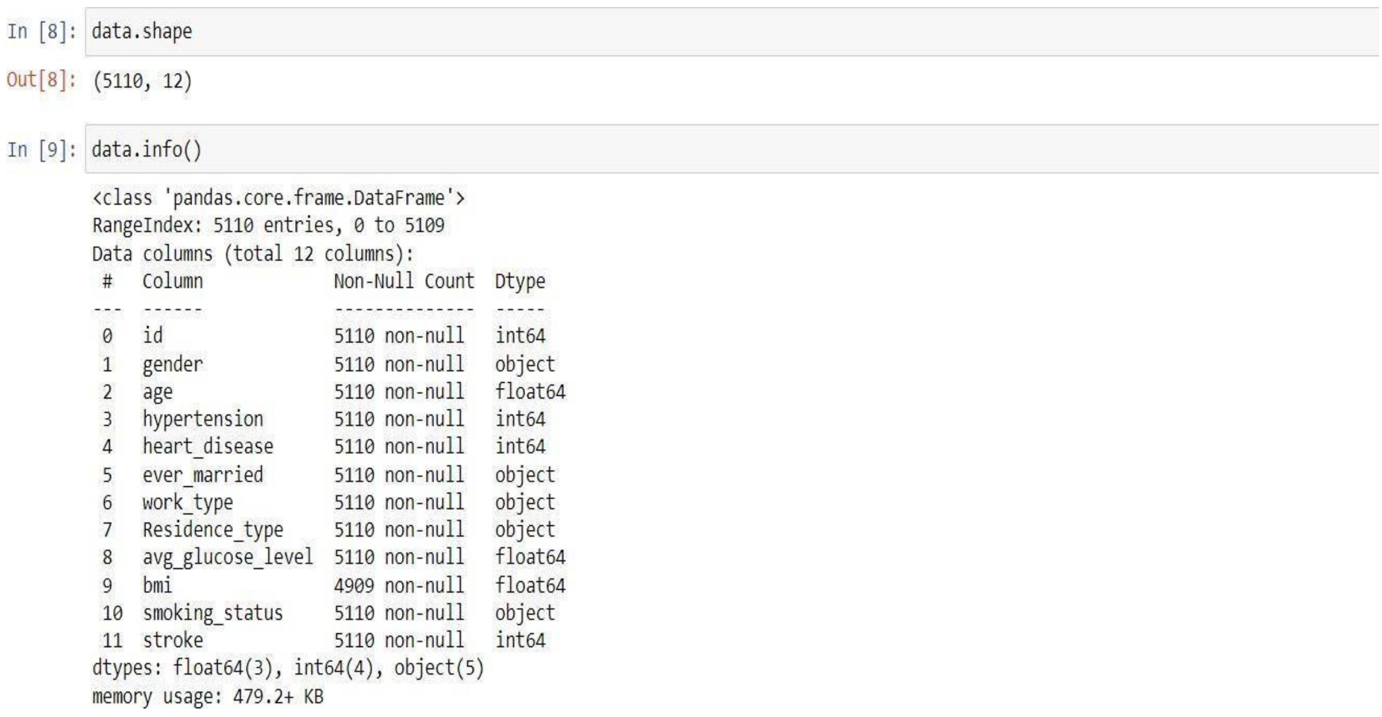


```
In [6]: data=pd.read_csv(r'C:\Users\HP\Desktop\trail\Stroke-Risk-Prediction-using-Machine-Learning-master\healthcare-dataset-stroke-data.

In [7]: data

Out[7]:
         id  gender   age  hypertension  heart_disease  ever_married      work_type  Residence_type  avg_glucose_level   bmi   smoking_status  stroke
   0   9046    Male  67.0             0              1           Yes        Private           Urban             228.69  36.6  formerly smoked       1
   1  51676  Female  61.0             0              0           Yes  Self-employed           Rural             202.21   NaN     never smoked       1
   2  31112    Male  80.0             0              1           Yes        Private           Rural             105.92  32.5     never smoked       1
   3  60182  Female  49.0             0              0           Yes        Private           Urban             171.23  34.4           smokes       1
   4   1665  Female  79.0             1              0           Yes  Self-employed           Rural             174.12  24.0     never smoked       1
 ...    ...     ...   ...           ...            ...           ...            ...             ...                ...   ...              ...     ...
5105  18234  Female  80.0             1              0           Yes        Private           Urban              83.75   NaN     never smoked       0
5106  44873  Female  81.0             0              0           Yes  Self-employed           Urban             125.20  40.0     never smoked       0
5107  19723  Female  35.0             0              0           Yes  Self-employed           Rural              82.99  30.6     never smoked       0
5108  37544    Male  51.0             0              0           Yes        Private           Rural             166.29  25.6  formerly smoked       0
5109  44679  Female  44.0             0              0           Yes       Govt_job           Urban              85.28  26.2          Unknown       0
```

Fig 4.2 Extracting and reading the datase

4.3 Feature variables

A feature is a measurable property of the object you’re trying to analyze. In datasets, features appear as columns. Features are the basic building blocks of datasets. The quality of the features in your dataset has a major impact on the quality of the insights you will gain when you use that dataset for machine learning. The target variable of a dataset is the feature of a dataset about which you want to gain a deeper understanding. Asupervised machine learning algorithm uses historical data to learn patterns and uncover relationships between other features of your dataset and the target. Without a labeled target, supervised machine learning algorithms would be unable to map available data to outcomes.



```
In [8]: data.shape
Out[8]: (5110, 12)

In [9]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5110 entries, 0 to 5109
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 5110 non-null   int64
 1   gender             5110 non-null   object
 2   age                5110 non-null   float64
 3   hypertension       5110 non-null   int64
 4   heart_disease      5110 non-null   int64
 5   ever_married       5110 non-null   object
 6   work_type          5110 non-null   object
 7   Residence_type     5110 non-null   object
 8   avg_glucose_level  5110 non-null   float64
 9   bmi                4909 non-null   float64
 10  smoking_status     5110 non-null   object
 11  stroke             5110 non-null   int64
dtypes: float64(3), int64(4), object(5)
memory usage: 479.2+ KB
```

Fig 4.3 Feature variables

4.4 Data outlier removation

Sometimes a dataset can contain extreme values that are outside the range of what is expected and unlike the other data. These are called outliers and often machine learning modeling and model skill in general can be improved by understanding and even removing these outlier values.

Outliers can have many causes, such as:

- Measurement or input error.
- Data corruption.
- True outlier observation (e.g. Michael Jordan in basketball).

There is no precise way to define and identify outliers in general because of the specifics of each dataset. Instead, you, or a domain expert, must interpret the raw observations and decide whether a value is an outlier or not.

Outlier Removation

```
In [18]: from matplotlib.pyplot import figure
         figure(num=None, figsize=(8, 6), dpi=800, facecolor='w', edgecolor='k')

Out[18]: <Figure size 6400x4800 with 0 Axes>

         <Figure size 6400x4800 with 0 Axes>

In [19]: data.plot(kind='box')
         plt.show()
```

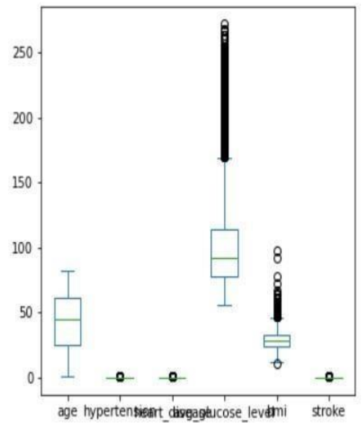
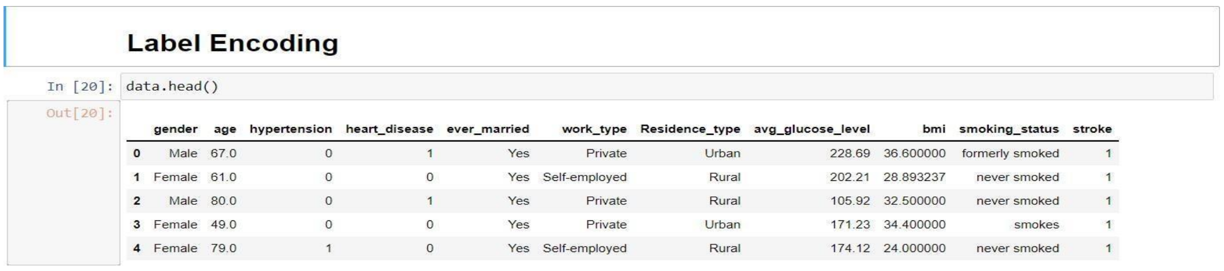


Fig 4.4 Outlier removation

4.5 Label Encoding

Label encoding encodes the string literals in the dataset into integer values for the machine to understand them. As the machine is usually trained in numbers, the strings have to be converted into integers. There are five columns in the collected dataset that have strings as their data type. On performing label encoding, all the strings get encoded, and the entire dataset becomes a combination of numerals.

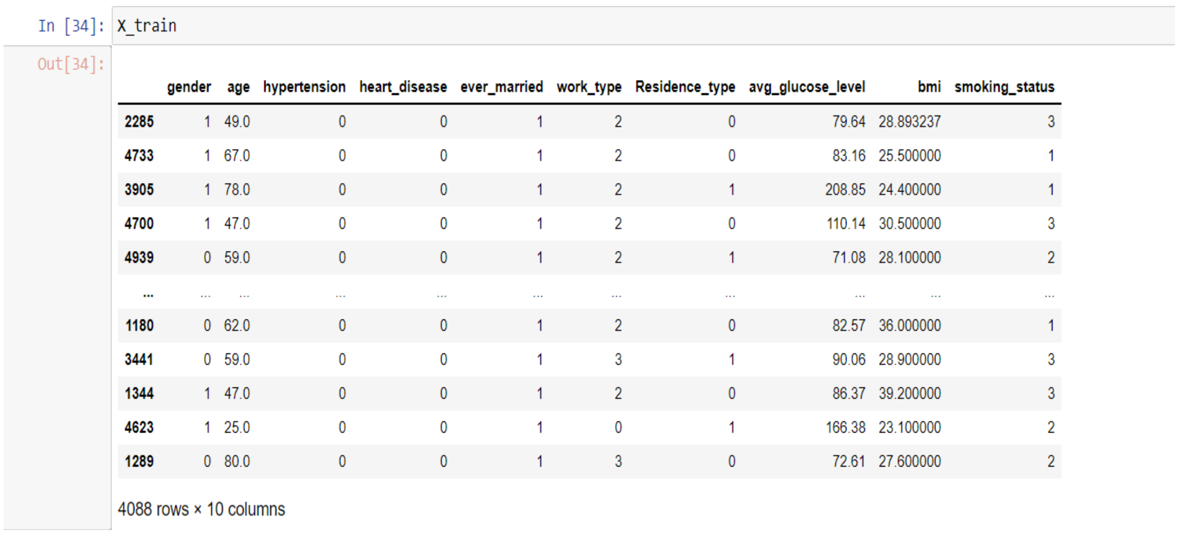


Label Encoding

```
In [20]: data.head()
Out[20]:
```

	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	Male	67.0	0	1	Yes	Private	Urban	228.69	36.600000	formerly smoked	1
1	Female	61.0	0	0	Yes	Self-employed	Rural	202.21	28.893237	never smoked	1
2	Male	80.0	0	1	Yes	Private	Rural	105.92	32.500000	never smoked	1
3	Female	49.0	0	0	Yes	Private	Urban	171.23	34.400000	smokes	1
4	Female	79.0	1	0	Yes	Self-employed	Rural	174.12	24.000000	never smoked	1

Fig 4.5(a) before label encoding



```
In [34]: X_train
Out[34]:
```

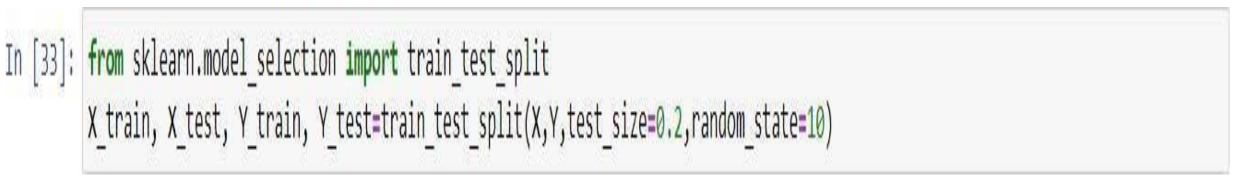
	gender	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status
2285	1	49.0	0	0	1	2	0	79.64	28.893237	3
4733	1	67.0	0	0	1	2	0	83.16	25.500000	1
3905	1	78.0	0	0	1	2	1	208.85	24.400000	1
4700	1	47.0	0	0	1	2	0	110.14	30.500000	3
4939	0	59.0	0	0	1	2	1	71.08	28.100000	2
...	...	...	...	...	...	...	...	...	...	...
1180	0	62.0	0	0	1	2	0	82.57	36.000000	1
3441	0	59.0	0	0	1	3	1	90.06	28.900000	3
1344	1	47.0	0	0	1	2	0	86.37	39.200000	3
4623	1	25.0	0	0	1	0	1	166.38	23.100000	2
1289	0	80.0	0	0	1	3	0	72.61	27.600000	2

4088 rows × 10 columns

Fig 4.5(b) after label encoding

4.6 Splitting the data into training and testing (80% - 20%split)

A very common issue when training a model is overfitting. This phenomenon occurs when a model performs really well on the data that we used to train it but it fails to generalize well to new, unseen data points. Typically, the higher the complexity of a model the higher the chance that it will be overfitted. On the other hand, underfitting occurs when the model has poor performance even on the data that was used to train it. Creating different data samples for training and testing the model is the most common approach that can be used to identify these sorts of issues. In this way, we can use the training set for training our model and then treat the testing set as a collection of data points that will help us evaluate whether the model can generalize well to new, unseen data. The simplest way to split the modelling dataset into training and testing sets is to assign 2/3 data points to the former and the remaining one-third to the latter. Therefore, we train the model using the training set and then apply the model to the test set. In this way, we can evaluate the performance of our model.

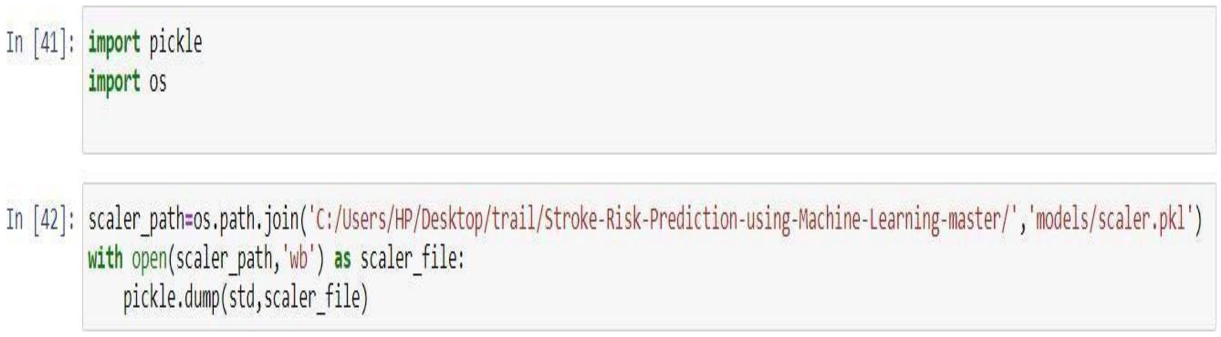


```
In [33]: from sklearn.model_selection import train_test_split
         X_train, X_test, Y_train, Y_test=train_test_split(X,Y,test_size=0.2,random_state=10)
```

Fig 4.6 Splitting the data into training and testing

4.7 Scalar object saving

The path parameters are either *strings or bytes* . These functions here are used for different purposes such as for merging, normalizing and retrieving path names in python . All of these functions accept either only bytes or only string objects as their parameters. The resultis an object of the same type, if a path or file name is returned
os.path.join() method in Python join one or more path components intelligently. This method concatenates various path components with exactly one directory separator (‘/’)following each non-empty part except the last path component. If the last path component to be joined is empty then a directory separator (‘/’) is put at the end.



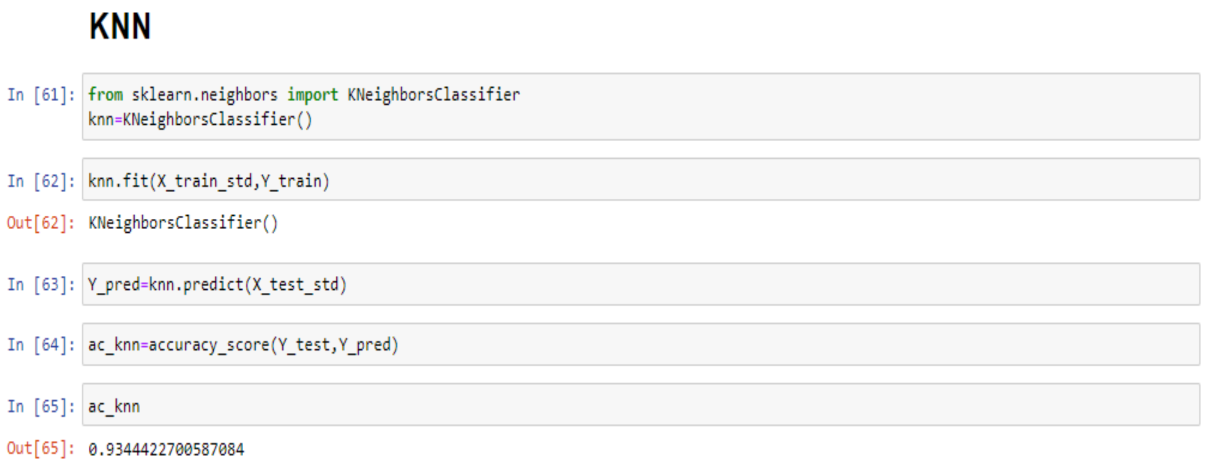
```
In [41]: import pickle
         import os

In [42]: scaler_path=os.path.join('C:/Users/HP/Desktop/trail/Stroke-Risk-Prediction-using-Machine-Learning-master/','models/scaler.pkl')
         with open(scaler_path,'wb') as scaler_file:
             pickle.dump(std,scaler_file)
```

Fig 4.7 Saving the scalar object

4.8 Training Algorithms

KNN

It stores the dataset, and at the time of classification, it acts on the dataset. The working principle of KNN is to find similarities between the new case (or data) and available data and then map the new case into the category that is most similar to the available categories.



KNN

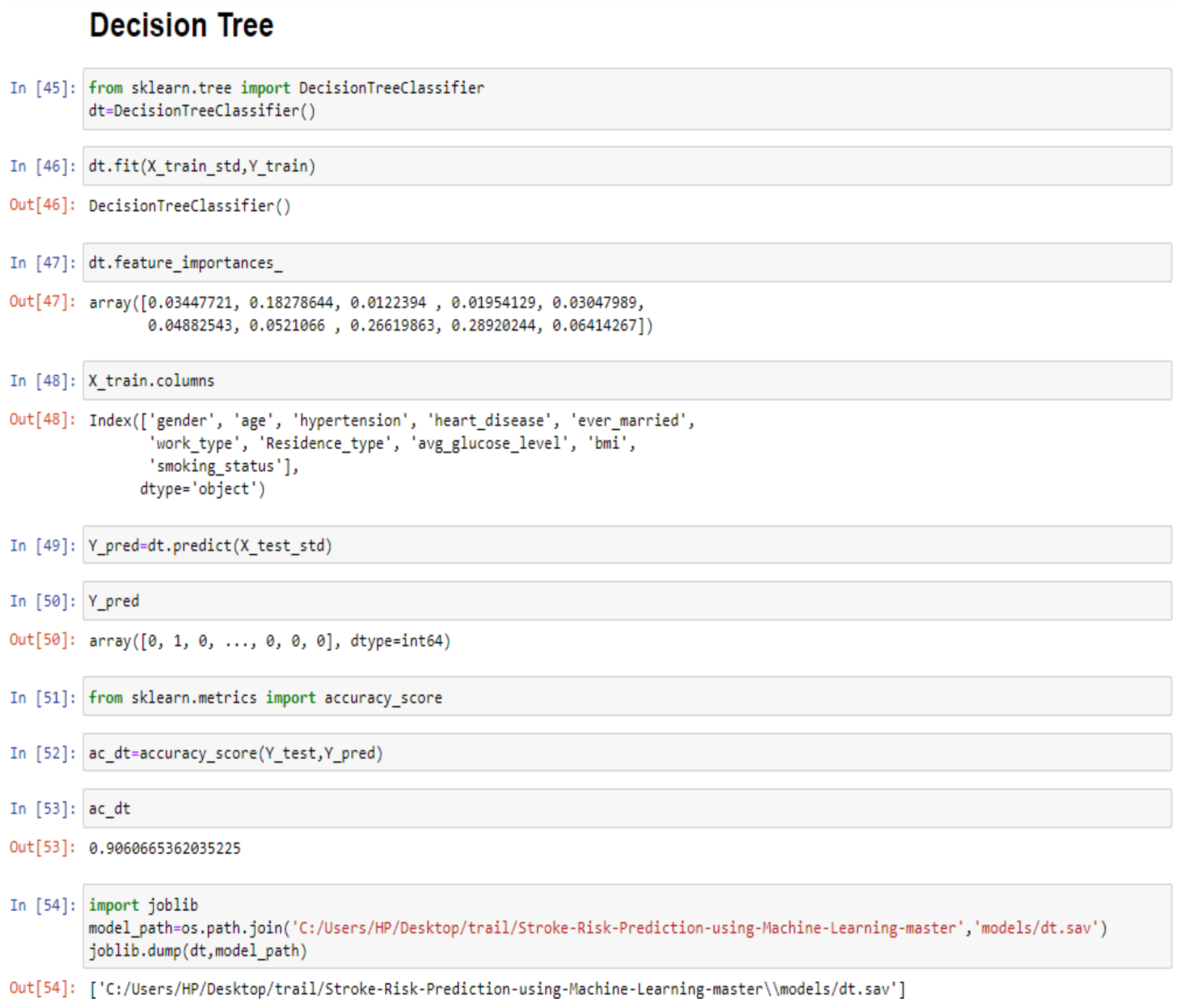
```
In [61]: from sklearn.neighbors import KNeighborsClassifier
         knn=KNeighborsClassifier()

In [62]: knn.fit(X_train_std,Y_train)

Out[62]: KNeighborsClassifier()

In [63]: Y_pred=knn.predict(X_test_std)

In [64]: ac_knn=accuracy_score(Y_test,Y_pred)

In [65]: ac_knn

Out[65]: 0.9344422700587084
```

Fig 4.8 KNN

4.9 Decision Tree

Decision Tree classification is to solve both Regression and classification problems. This algorithm is a supervised learning method wherein the input variables already have their corresponding output variable. It is a tree-like structure. In this algorithm, the data continuously splits according to a particular parameter. A decision tree has two parts:

Decision Node and Leaf node. The data splits at the former node, and the latter is the node that gives the outcome.



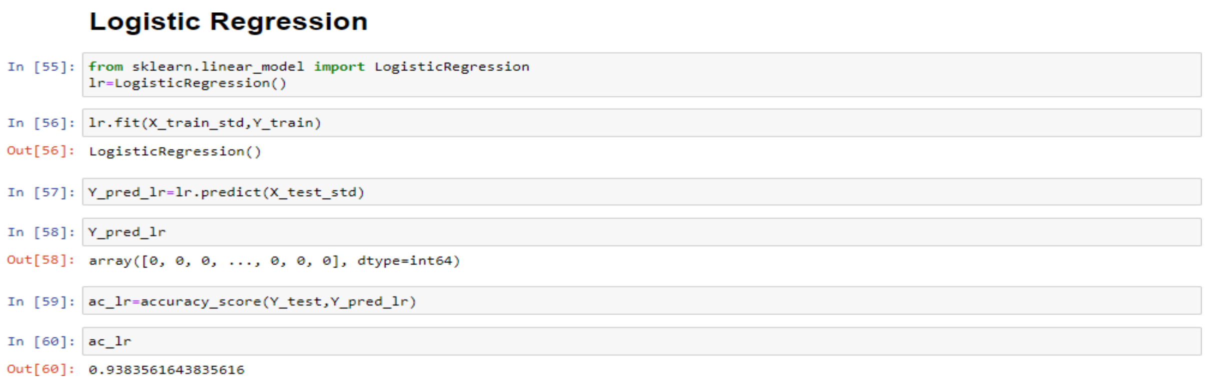
Decision Tree

```
In [45]: from sklearn.tree import DecisionTreeClassifier
         dt=DecisionTreeClassifier()

In [46]: dt.fit(X_train_std,Y_train)

Out[46]: DecisionTreeClassifier()

In [47]: dt.feature_importances_

Out[47]: array([0.03447721, 0.18278644, 0.0122394 , 0.01954129, 0.03047989,
                0.04882543, 0.0521066 , 0.26619863, 0.28920244, 0.06414267])

In [48]: X_train.columns

Out[48]: Index(['gender', 'age', 'hypertension', 'heart_disease', 'ever_married',
                'work_type', 'Residence_type', 'avg_glucose_level', 'bmi',
                'smoking_status'],
               dtype='object')

In [49]: Y_pred=dt.predict(X_test_std)

In [50]: Y_pred

Out[50]: array([0, 1, 0, ..., 0, 0, 0], dtype=int64)

In [51]: from sklearn.metrics import accuracy_score

In [52]: ac_dt=accuracy_score(Y_test,Y_pred)

In [53]: ac_dt

Out[53]: 0.9060665362035225

In [54]: import joblib
         model_path=os.path.join('C:/Users/HP/Desktop/trail/Stroke-Risk-Prediction-using-Machine-Learning-master','models/dt.sav')
         joblib.dump(dt,model_path)

Out[54]: ['C:/Users/HP/Desktop/trail/Stroke-Risk-Prediction-using-Machine-Learning-master\\models/dt.sav']
```

fig 4.9 Decision Tree

4.10 Logistic Regression

Logistic Regression is a supervised learning algorithm used for predicting the probability of the output variable. This algorithm is the best fit when the output variable has binary values (0 or 1). As the output attribute in the dataset has only two possible values.



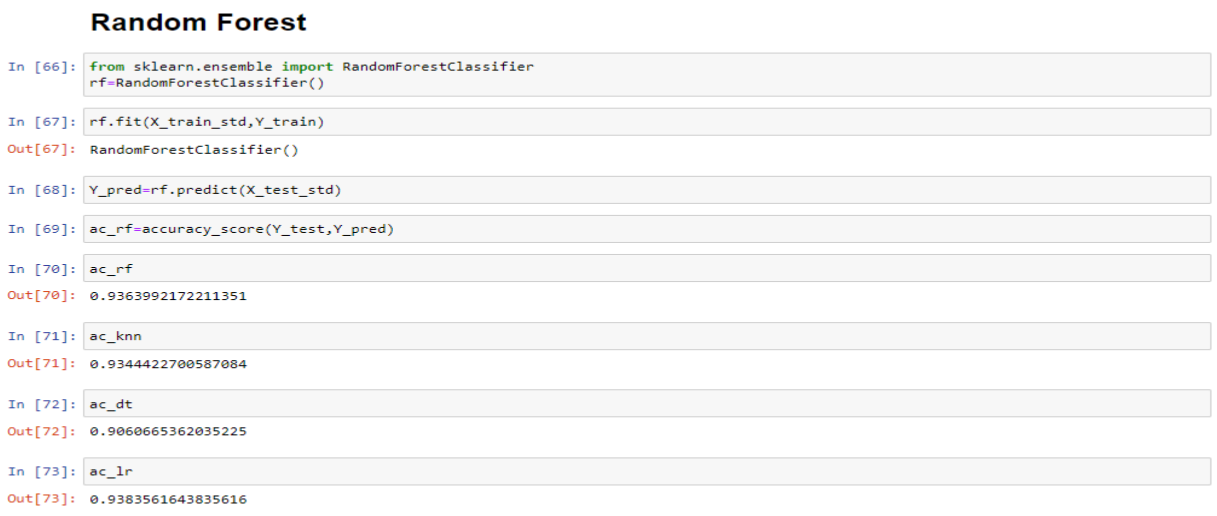
Logistic Regression

```
In [55]: from sklearn.linear_model import LogisticRegression
         lr=LogisticRegression()

In [56]: lr.fit(X_train_std,Y_train)
Out[56]: LogisticRegression()

In [57]: Y_pred_lr=lr.predict(X_test_std)

In [58]: Y_pred_lr
Out[58]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)

In [59]: ac_lr=accuracy_score(Y_test,Y_pred_lr)

In [60]: ac_lr
Out[60]: 0.9383561643835616
```

Fig 4.10 Logistic Regression

4.11 Random Forest

Random Forests are composed of multiple independent decision trees trained independently on a random subset of data.These trees are generated at the time of training, and the outputsare obtained from each decision tree.For the final prediction from this algorithm, a method called "voting" takes place. This method means that each decision tree votes for an outputclass (in this case, the two classes are: 'stroke' and 'no stroke'). The random forest chooses the class withthe maximum number of votes as the final prediction.



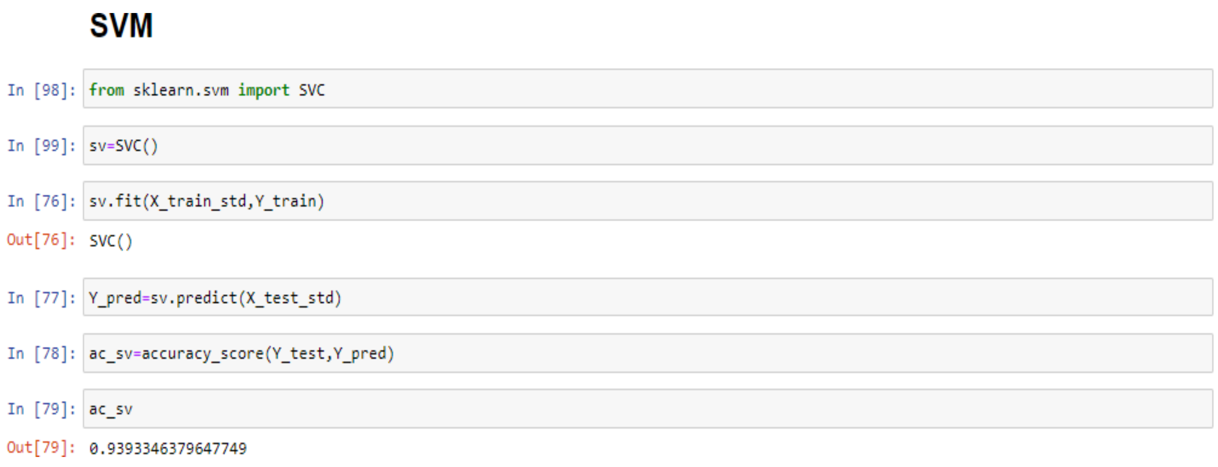
Random Forest

```
In [66]: from sklearn.ensemble import RandomForestClassifier
         rf=RandomForestClassifier()

In [67]: rf.fit(X_train_std,Y_train)
Out[67]: RandomForestClassifier()

In [68]: Y_pred=rf.predict(X_test_std)

In [69]: ac_rf=accuracy_score(Y_test,Y_pred)

In [70]: ac_rf
Out[70]: 0.9363992172211351

In [71]: ac_knn
Out[71]: 0.9344422700587084

In [72]: ac_dt
Out[72]: 0.9060665362035225

In [73]: ac_lr
Out[73]: 0.9383561643835616
```

Fig 4.11 Random Forest

4.12 SVM

It is a supervised learning technique that can be associated with learning algorithms to analyze the data for both classification and regression. Support Vector Machine (SVM) scales relatively well to high dimensional data.SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors.

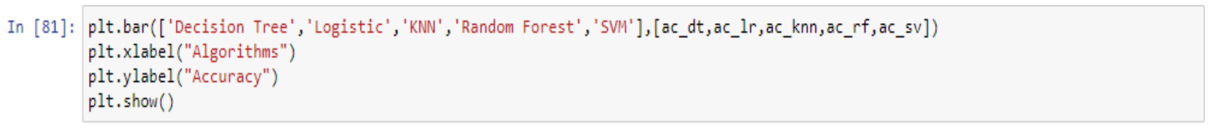


```
SVM

In [98]: from sklearn.svm import SVC

In [99]: sv=SVC()

In [76]: sv.fit(X_train_std,Y_train)
Out[76]: SVC()

In [77]: Y_pred=sv.predict(X_test_std)

In [78]: ac_sv=accuracy_score(Y_test,Y_pred)

In [79]: ac_sv
Out[79]: 0.9393346379647749
```

Fig 4.12 SVM

4.13 Accuracy Comparision of Algorithms

Logistic regression algorithm performs best with an accuracy of 93.835%. The accuracy of Random Forest, KNN, Decision Tree, SVM are 93.639%, 93.44%, 90.60% and 93.93% respectively.



```
In [81]: plt.bar(['Decision Tree','Logistic','KNN','Random Forest','SVM'],[ac_dt,ac_lr,ac_knn,ac_rf,ac_sv])
         plt.xlabel("Algorithms")
         plt.ylabel("Accuracy")
         plt.show()
```

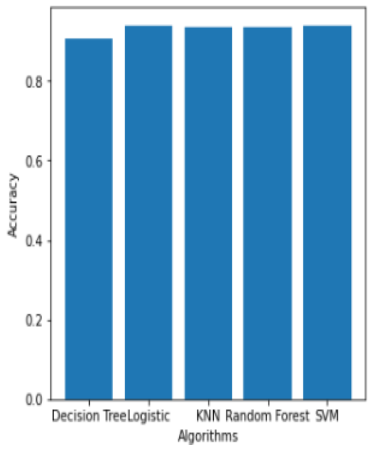
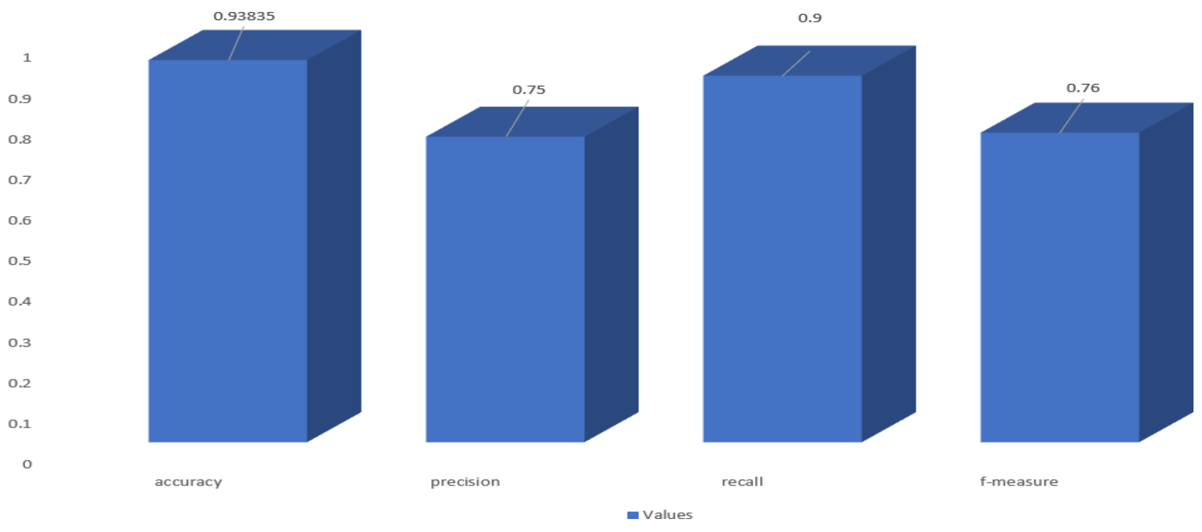
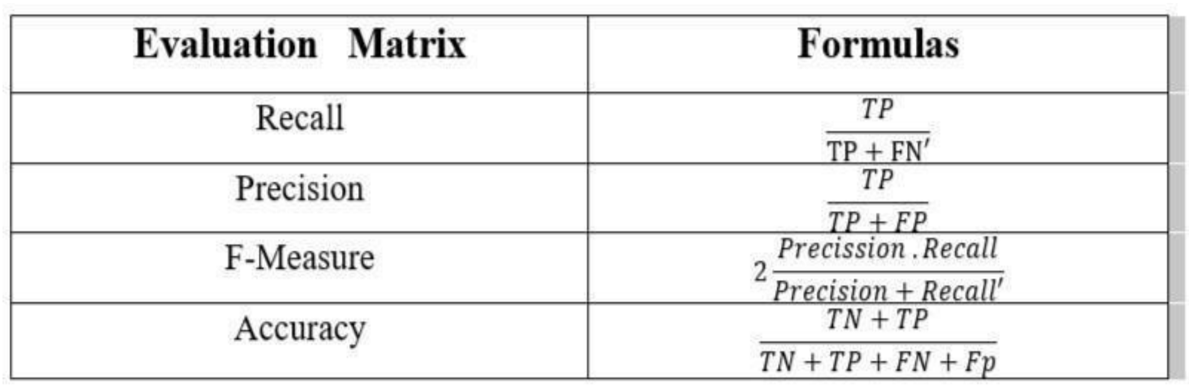


Fig 4.13 Accuracy Comparision



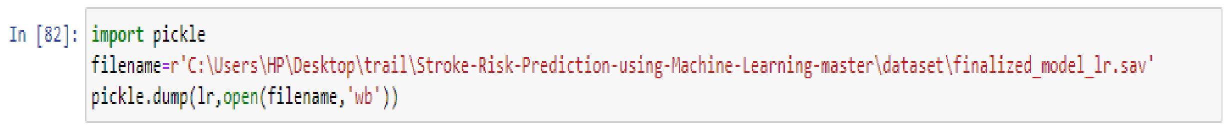


Evaluation Matrix	Formulas
Recall	$\frac{TP}{TP+FN}$'
Precision	$\frac{TP}{TP+FP}$
F-Measure	$2\frac{Precission\,.Recall}{Precision+Recall}$'
Accuracy	$\frac{TN+TP}{TN+TP+FN+Fp}$

4.14 Generating Pickle File

The pickle module allows you to pickle a file using de-serialization which means simply breaking down an object into its constituting components.

The pickle module keeps track of the objects it has already serialized, so that later references to the same object won't be serialized again, thus allowing for faster execution time. Allows saving model in very little time. Good For small models with fewer parameters like the one we used.



```
In [82]: import pickle
         filename=r'C:\Users\HP\Desktop\trail\Stroke-Risk-Prediction-using-Machine-Learning-master\dataset\finalized_model_lr.sav'
         pickle.dump(lr,open(filename,'wb'))
```

Fig 4.14 Pickle File Generation

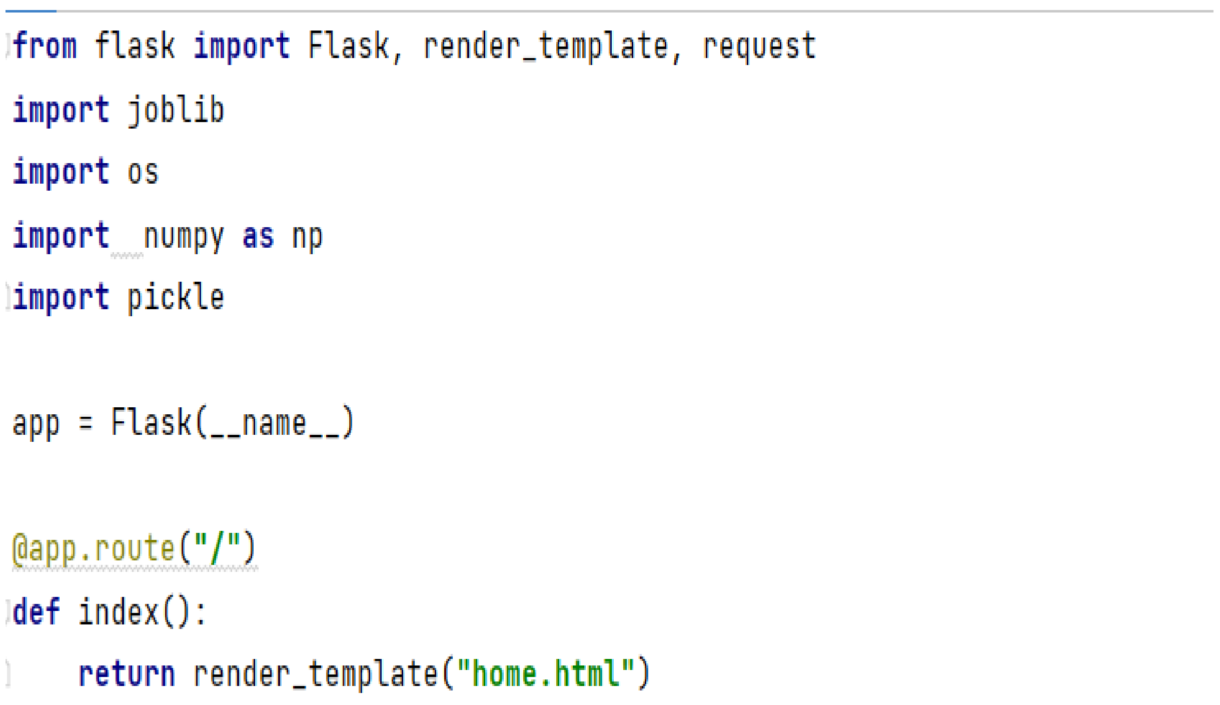
4.15 The libraries imported in PyCharm

PyCharm supports Flask framework development. You can easily create a new Flask project by creating new project through welcome screen. You can set the project's location and the virtual environment and choose a template language and where the templates will be located.

Joblib has no mandatory dependencies besides Python. Joblib can efficiently dump and load numpy arrays but does not require numpy to be installed.

NumPy stands for numerical python is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. It is an open-Source project. In Python we have lists that serve the purpose of arrays, but they are slow toprocess. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The pickle module implements a fundamental, but powerful algorithm for serializing and de-serializing a Python object structure



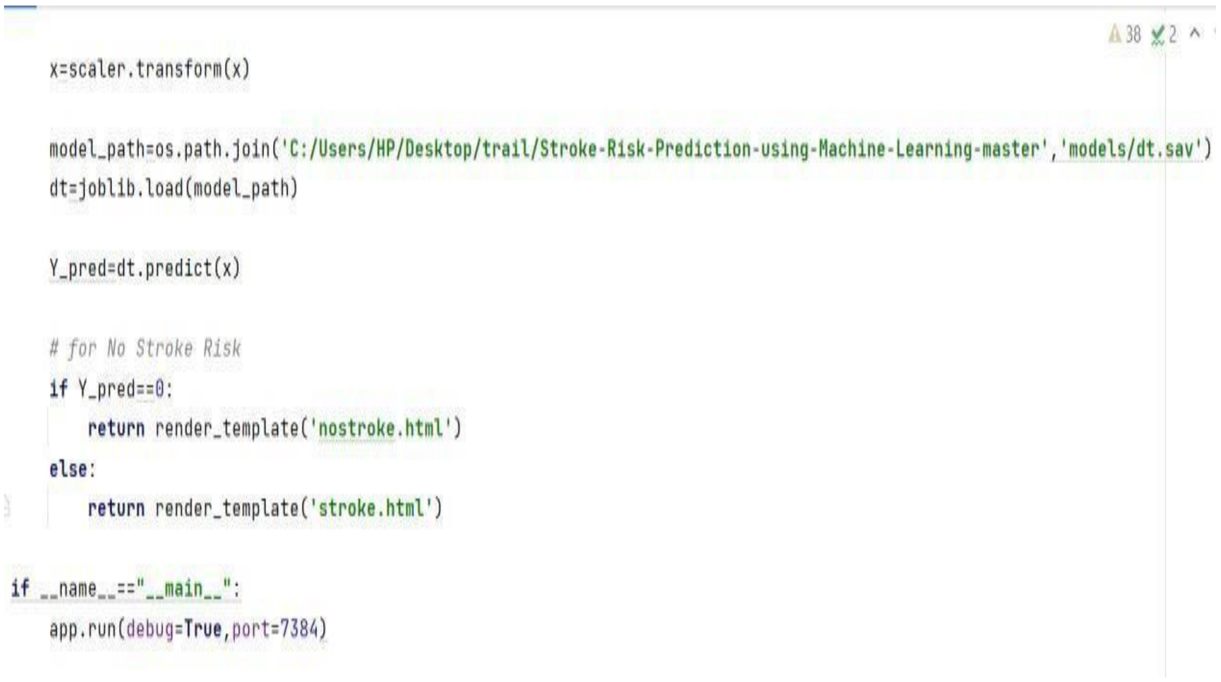
```
from flask import Flask, render_template, request
import joblib
import os
import  numpy as np
import pickle


app = Flask(__name__)


@app.route("/")
def index():
    return render_template("home.html")
```

Fig 4.15 The libraries imported in PyCharm

4.16 The logic in PyCharm

This code predicts whether a person has a risk of smoke or no smoke

.

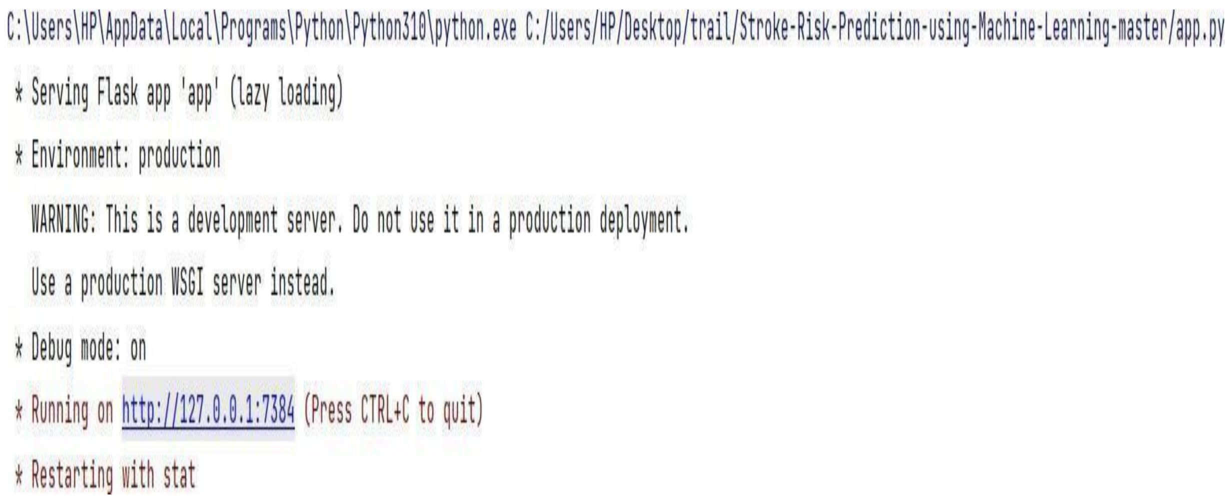


```
    x=scaler.transform(x)

    model_path=os.path.join('C:/Users/HP/Desktop/trail/Stroke-Risk-Prediction-using-Machine-Learning-master','models/dt.sav')
    dt=joblib.load(model_path)

    Y_pred=dt.predict(x)

    # for No Stroke Risk
    if Y_pred==0:
        return render_template('nostroke.html')
    else:
        return render_template('stroke.html')

if __name__=="__main__":
    app.run(debug=True,port=7384)
```

Fig 4.16 The logic in PyCharm

4.17 The output from pycharm

In order to open html page, click on the http link (CTRL+C to quit) that’s available after running the codeas shown in the fig below. After clinking on the given link it takes user to a page where user can give input.

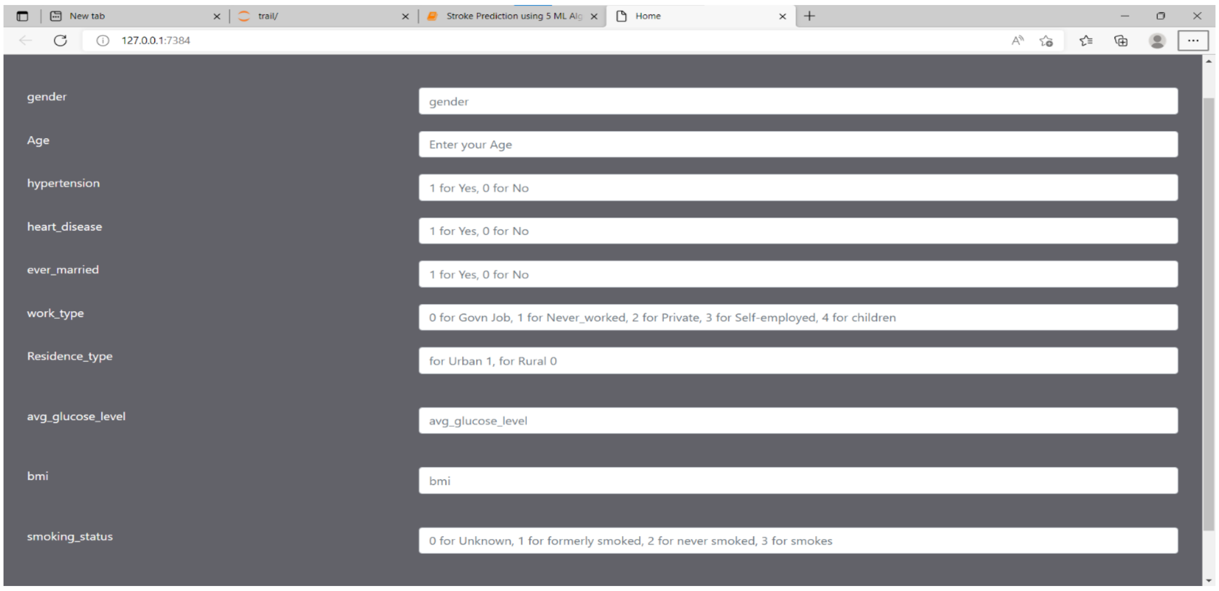


```
C:\Users\HP\AppData\Local\Programs\Python\Python310\python.exe C:/Users/HP/Desktop/trail/Stroke-Risk-Prediction-using-Machine-Learning-master/app.py
 * Serving Flask app 'app' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:7384 (Press CTRL+C to quit)
 * Restarting with stat
```

Fig 4.17 the output from pycharm

4.18 The html page

This page takes input from the user as gender, age, hypertension, bmi, heart_disease, work_type, avg_glucose_level, smoking_status, Residence_type, ever_married and predicts the output.

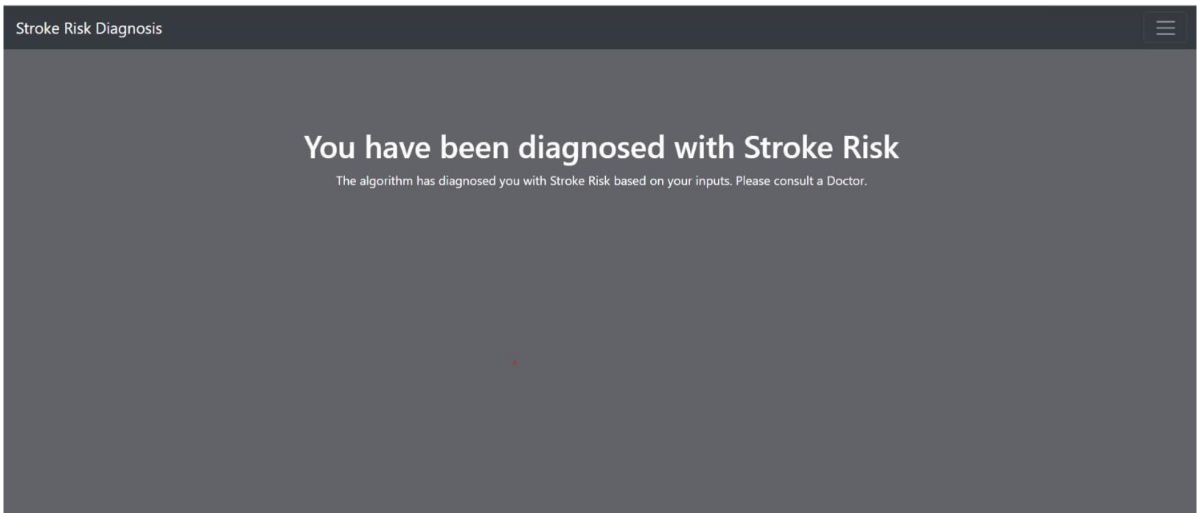


gender	gender
Age	Enter your Age
hypertension	1 for Yes, 0 for No
heart_disease	1 for Yes, 0 for No
ever_married	1 for Yes, 0 for No
work_type	0 for Govn Job, 1 for Never_worked, 2 for Private, 3 for Self-employed, 4 for children
Residence_type	for Urban 1, for Rural 0
avg_glucose_level	avg_glucose_level
bmi	bmi
smoking_status	0 for Unknown, 1 for formerly smoked, 2 for never smoked, 3 for smokes

Fig 4.18 The html page

4.19 The predicted stroke risk output

The output can be displayed as a response to the given input as shown in figure if a person has the stroke risk or not.



Stroke Risk Diagnosis

You have been diagnosed with Stroke Risk

The algorithm has diagnosed you with Stroke Risk based on your inputs. Please consult a Doctor.

Fig 4.19 The predicted stroke risk output

CHAPTER 5

Conclusion And Future Scope

Conclusion

Stroke is a critical medical condition that should be treated before it worsens. Building a machine learning model can help in the early prediction of stroke and reduce the severe impact of the future.

The stroke prediction Kaggle dataset was used to perform the study. There were 5110 rows and 12 columns in this dataset. Data preprocessing is done to remove unwanted noise andoutliers from the dataset that could lead the model to depart from its intended training. After the training is done using Random Forest, Decision Tree, KNN, Logistic Regression, SVM algorithms. Testing is done by building the web page using simple HTML code. This application has an input form that will take the entered input values from the user to predict the occurrence of stroke. For evaluating the performance of models, we have used the confusion matrix to calculate accuracy, precision, recall, and f-measure.

The performance of various machine learning algorithms in successfully predicting stroke based on multiple physiological attributes. Out of Random Forest, Decision Tree, KNN, Logistic Regression, SVM algorithms chosen, Logistic regression algorithm performs best with an accuracy of 93.835%, precision of 75%, recall of 90%, f-measure of 76%.

Future scope:

The future scope of this study is that using a larger dataset and machine learning models, such as AdaBoost, SVM, and Bagging, the framework models may be enhanced. This will enhance the dependability of the framework and the framework's presentation. In exchange for just providing some basic information, the machine learning architecture may help the general public in determining the likelihood of a stroke occurring in an adult patient. In an ideal world, it would help patients obtain early treatment for strokes and rebuild their lives after the event.

References:

[1] S. Cheon, J. Kim, and J. Lim, “The use of deep learning to predict stroke patient mortality,” *International Journal of Environmental Research and Public Health*, vol. 16, no. 11,2019.View at: Publisher Site | Google Scholar

[2] M. S. Zulfiker, N. Kabir, A. A. Biswas, P. Chakraborty, and M. M. Rahman, “Predicting students’ performance of the private universities of Bangladesh using machine learning approaches,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3,2020.View at: Publisher Site | Google Scholar

[3] Tasfia Ismail Shoily, Tajul Islam, , Sumaiya Jannat and Sharmin Akter Tanna "Detection ofstroke using machine learning algorithms", 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, July 2019

[4] JoonNyungHeo, Jihoon G. Yoon , Hyungjong Park, Young Dae Kim, Hyo Suk Nam and JiHoe Heo. "Stroke prediction in acute stroke", Stroke. 2019;50:1263-1265, AHA Journal, 20 Mar2019.

[5] Lin, C.-H., Hsu, K.-C., Johnson, K. R., Fann, Y. C., Tsai, C.-H., Sun, Y., et al. (2020). Evaluation of Machine Learning Methods to Stroke Outcome Prediction Using a NationwideDisease Registry. Comp. Methods Programs Biomed. 190, 105381. doi:10.1016/j.cmpb.2020.105381

[6] Ge, Y., Wang, Q., Wang, L., Wu, H., Peng, C., Wang, J., et al. (2019). Predicting post- strokePneumonia Using Deep Neural Network Approaches. Int. J. Med. Inform. 132, 103986. doi:10.1016/j.ijmedinf.2019.103986

Student Information	Student Address	Photographs
SHASHIDHAR GOWDA S 4MC18EC062 Shashidhargowda18@gmail.com Mobile : 8722055065	M Hosakoppalu, H N pura road, Hassan	
SHREYAS B C 4MC18EC064 Shreyasbc19@gmail.com Mobile : 9108428952	Byrapura(village at post) Alur (tq) Hassn (district)	
SPOORTHI K R 4MC18EC070 Spoorthikr1@gmail.com Mobile : 8105232760	Kasturi Layout,Near Gorur Bypass road, Behind HP petrol Bunk, Bittagowdanahalli, hassan	
SWATHI H AIRANI 4MC18EC074 Swathihairani5102000@gmail.com Mobile : 9380432916	Hig 209, Aashirvad nilaya, KHB colony, 6th ward, 4th main, near lakshminarayama temple, Sathyamangala, Hassan	